Akenti – A Distributed Access Control System

Srilekha S. Mudumbai (SSMudumbai@lbl.gov), *William Johnston* (wejohnston@lbl.gov),
Mary R. Thompson (mrthompson@lbl.gov), *Abdeliah Essiari* (aes@lbl.gov)
Imaging and Distributed Collaboration Group
Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Abstract: DOE scientific resources – instruments, data, and collaborations – that are accessed via open networks require protection against unauthorized use. Akenti is designed to provide a flexible, easily managed mechanism, which strongly controls access to distributed resources, by widely distributed users.

1.0 Introduction

The people with authority to grant access to resources (stakeholders) may be both physically and organizationally remote from the resource, especially in the scientific environment. Akenti enables these stakeholders to remotely and securely create and distribute instructions authorizing access to their resources.

Access control is a means for enforcing an authorization policy. In a client-server architecture, the clients (on behalf of users) attempt to access resources that are controlled by servers. A priori authorization decisions govern which clients may access which servers for what purposes and under what conditions. These decisions are reflected in an access control policy. Akenti makes access control decisions based on a set of digitally signed documents that represent the authorization instructions. Existing public-key infrastructure and security systems provide confidentiality, message integrity, and user identity authentication, during and after the access decision process. (Fig. 1)

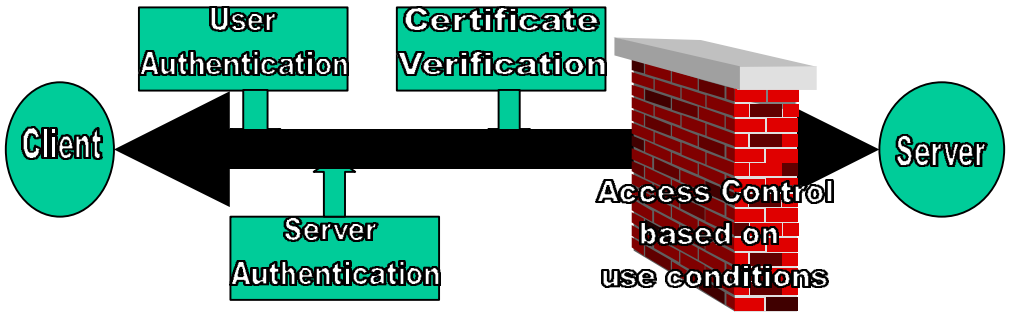
2.0 Access Control

A "resource" may be information, processing or communication capabilities, physical systems, etc. "Access" can mean the ability to obtain information from the resource (as in "read" access), to modifying the resource (as in "write" access), or cause that resource to perform certain functions (as in changing instrument control set points). Remote access to a resource is typically provided by a network-based server acting as a proxy for the resource. A user is a person attempting to gain access to the resource via a client. The client, therefore, has to participate in a series of authentication and verification steps before gaining access to the resource:

- In an initial two-way process, the client authenticates the server, after which it will accept information from that server. Following this, the server authenticates the client by checking the user's identity credentials.
- There may be a set of use-conditions imposed on the resource by the stakeholders in the form of digitally signed documents ("certificates").
- If the user can satisfy the use-conditions, then the client is permitted access and actions (i.e. read, modify etc.) on the resource.

3.0 Authorization

Authorization is the granting of rights, by the owner or controller of a resource, for others to access that resource. This authorization is enforced at different levels by access control.



Access Control Model (Fig 1)

Initially identity authorization is required in gaining access to the server. This is accomplished by checking user's identity in the form of Public-Key Infrastructure (PKI) X.509 certificate. If the user has an identity certificate issued by a Certificate Authority trusted by the server, the initial authorization succeeds.

A second-level of authorization is required to verify that the user satisfies the use-conditions that are required by the stakeholders ("owners" and policymakers) in order to access a resource. The stakeholders are responsible for establishing policy at the resource level. The use-condition issuer, to whom the policy makers delegate access control authority, is responsible for issuing use-conditions for that resource. The use-condition certificates also define the acceptable attribute certificate issuers.

Use-condition certificates: These are signed documents, remotely created and manipulated by resource stakeholders, that specify the conditions that must be met by a user wishing to access the resource. They include

- Combinations of required attributes and values
- Name of the resource
- Permitted actions
- Identities to be trusted to issue related attribute certificates
- Certificate Authorities (CA) to be trusted to verify the user and issuer identities
- Signature of the use-condition issuer.

Attribute certificates: These are signed documents, stored in trusted servers that certify that a user possesses a specific attribute, (for example, membership in a named group, has completed a certain training course, etc.) They include

- Attribute name
- Value
- Auxiliary information (e.g. time interval)
- Subject (User) and its trusted CA
- Issuer and its trusted CA
- Signature of the attribute certificate issuer.

Certificate Content:

```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: CN=IDC6-CA, OU=ICSD, O=Lawrence Berkeley National Laboratory, C=US
        Validity:
            Not Before: Fri Aug 29 11:25:05 1997
            Not  After: Sat Feb 20 10:25:05 1999
        Subject: CN=Srilekha Mudumbai, OU=ICSD, O=Lawrence Berkeley National Laboratory,
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:9c:23:9e:55:bf:50:9c:99:75:7d:02:fe:77:3e:b9:ec:90:
                    90:8c:a5:e9:0f:99:1a:db:1f:c7:db:9d:c2:36:02:c4:c9:bc:
                    14:94:9e:06:a0:a5:ba:26:0f:3d:2e:b3:d5:b7:a7:cc:19:02:
                    d4:1c:d5:09:c4:67:f0:f2:e0:bd:9b
                Public Exponent: 65537 (0x10001)
        Extensions:
            Identifier: Certificate Type
                Critical: no
                Certified Usage:
                    SSL Client
            Identifier: Authority Key Identifier
                Critical: no
                Key Identifier:
                    09:07:1d:ab:52:ef:c1:5a:6b:33:b9:0b:94:f2:e5:ed:f9:96:
                    e0:fb
    Signature:
        Algorithm: PKCS #1 MD5 With RSA Encryption
        Signature:
            91:52:28:a1:48:97:5c:e8:51:3c:c1:83:4c:7d:b4:bb:09:cd:ae:ec:82:
            e8:71:2d:2a:f8:73:a8:55:fd:f1:50:94:ee:37:1b:5c:10:da:47:23:be:
            0e:44:81:c0:3e:1d:65:9d:2a:1e:6f:05:16:d6:00:46:27:78:57:d6:58:
            9e:7f:5d:b1:c1:4e:12:1b:39:2a:53:2a:94:a6:2b:1b:a6:e6:ed:a6:e3:
            4a:9c:de:11:15:f6:e5:20:9c:d7:bc:ae:77:8c:12:bc:c0:4b:38:58:06:
            11:a0:01:c2:70:8b:b6:75:4d:0d:15:48:ab:c8:b6:4b:da:7b:3b:91:c3:
            06:ba
```

X.509 Certificate (Fig 2)

3.1 *Components of the Model*

Identity (X.509) certificates: PKI is used to generate and manipulate standard X.509 certificates (Fig. 2). These certificates associate an entity's name with a public-key, and binds them together through the digital signature of a Certificate Authority (CA). (The corresponding private-key is held secret by the user, and is used, among other things, to prove "ownership" of the corresponding public-key.)

Authority file: This is information is securely associated with each resource. This is relatively static information that specifies who can provide access information (i.e. defines the stakeholders), where to look for access-control information, and what CAs to trust.

During the second phase of authorization, use-condition certificates are searched for a list of attributes and values that the user has to satisfy. If there are corresponding attribute certificates available for the user, then the use-conditions are satisfied, and the user's client is allowed access. Contrary wise, access is denied.

4.0 Access Control Policy Model

Various policy models must be supported to accommodate different environments. Policy models differ in terms of

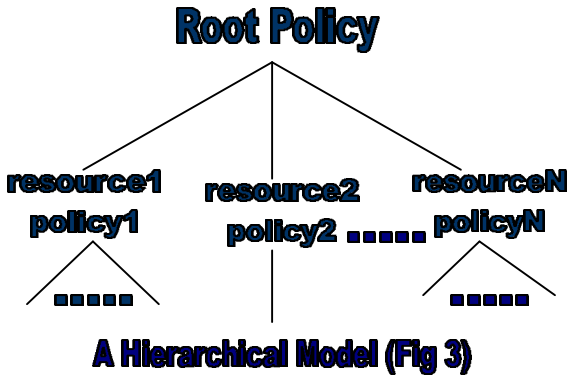
- The level at which the authorization decision is made (depending on whether the model supports delegation of authority, where decisions are made by stakeholders for specific resources, etc.);
- The ways in which users and/or resources are grouped together for purposes of common handling, and;
- The extent to which policies may be stated in terms of general rules.

A typical policy model is hierarchical (Fig. 3). Akenti's current model distinguishes between individual-based policies (stakeholders) and an overall policy. For example, a system administrator would be responsible for defining a root policy for the system, a Web administrator would define an overall policy for a Web directory hierarchy, etc. Root policy establishes trusted CA's and any global access restrictions. Individual-based policy is expressed in terms of a list of use-conditions for each resource, stating which users may perform what actions on that resource. Individual-based policies are the policies designed by stakeholders, for the resources that they own, as discussed above.

5.0 Implementation

5.1 ***Web Server***

Currently, Akenti implements a hierarchical policy model (Fig. 3) for the Apache Web Server that uses SSL (Secure Socket Layer) in order to provide confidentiality and integrity of communication after access is permitted. The "policy engine" is a module that that interfaces to the Apache server to provide Akenti's access control mechanism. This combination is being used to provide access control for some DOE scientific resources. The access control process involves five steps when a user attempts to access a resource (Fig. 4): (1) The client (user) authenticates the server; (2) the server authenticates the client by reference to a trusted CA; (3) the policy engine gathers remote use-condition certificates from stakeholder servers; (4) the attributes required by the use-conditions are verified by obtaining attribute certificates from trusted directories, and; (5) if the user satisfies all the requirements, a secure connection is established between the client and the server. Otherwise the policy engine denies access to the server's resource. LDAP (Lightweight Directory Access Protocol) directory servers are used to store information required for user's identity. X.509 Certificate Authorities are used to issue and digitally sign the identity certificates for the user. The policy engine talks to LDAP to verify issuers and user's identity as a part of access control. The identity of a subject (user or issuer) is established if the X.509 certificate is validated by any one of the trusted Certificate Authorities



A Hierarchical Model (Fig 3)

as specified by the authority files on the resource server.

5.2 *SPKM/GSS*

SPKM/GSS (Simple Public-Key GSS-API Mechanism) provides client-server developers with a secure communication mechanism based on public-key identity certificates. Akenti's policy engine will be incorporated into SPKM/GSS in order to provide access control as discussed above.

5.3 *Java*

The use-condition and attribute certificate generators are available via a graphical user interface. This is implemented using Java JDK1.1, Java Workshop and JDBC.

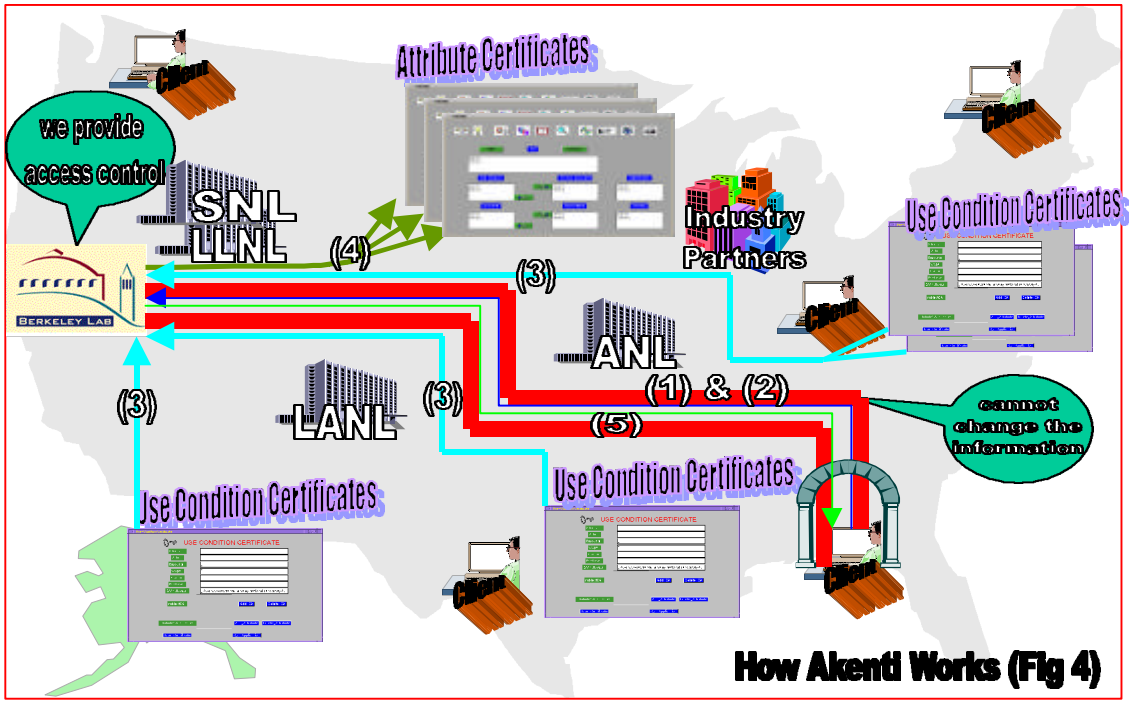
6.0 Future Work

Future work will concentrate on agent-based monitoring and management of the access control system using Java/KQML agents.

7.0 For more information

A detailed view of the project can be obtained from the following web sites

http://www-itg.lbl.gov/security
http://idcg-ca.lbl.gov



http://idcg-ds.lbl.gov/dshtml

An early deployment of Akenti is being used in support of the DOE2000, Diesel Engine Collaboratory. The web site is

(http://www-itg.lbl.gov/DieselCollab).

8.0 Acknowledgement

The work described above is supported by the U. S. Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office under contract DE-AC03-76SF00098 with the University of California.